UNCLASSIFIED

AD NUMBER

ADB043556

NEW LIMITATION CHANGE

TO

Approved for public release, distribution unlimited

FROM

Distribution authorized to U.S. Gov't. agencies and their contractors; Specific Authority; Nov 1979. Other requests shall be referred to Rome Air Development Center, Attn: DCLD, Griffiss AFB, NY 13441.

AUTHORITY

AFRL/IFOIP ltr, 1 Jun 2004

THIS PAGE IS UNCLASSIFIED

UNCLASSIFIED

AD NUMBER

ADB043556

NEW LIMITATION CHANGE

TO

Distribution authorized to U.S. Gov't. agencies and their contractors; Specific Authority; Nov 1979. Other requests shall be referred to Rome Air Development Center, Attn: DCLD. Griffiss AFB, NY 13441.

FROM

Distribution authorized to U.S. Gov't. agencies only; Test and Evaluation; Nov 1979. Other requests shall be referred to Rome Air Development Center, Attn: DCLD. Griffiss AFB, NY 13441.

AUTHORITY

RADC ltr, per DTIC Form 55

THIS PAGE IS UNCLASSIFIED

AD B043556

LEVEL II

RADC-TR-79-278
Final Technical Report
November 1979

16 KB/S DATA MODEM PARTITIONING

Harris Corporation

Fred C. Killmeyer
Daniel D. McRae

DDC
RECEIVED
JAN 29 1980
A

DDC FILE COPY

DISTRIBUTION LIMITED TO U.S. GOVERNMENT AGENCIES ONLY; TEST AND EVALUATION; Nov 1979 . OTHER REQUESTS FOR THIS DOCUMENT MUST BE REFERRED TO RADC (DCLD), GRIFFISS AFB NY 13441.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

80 1 28 093

RADC-TR-79-278 has been reviewed and is approved for publication.

APPROVED: Salvatore J. Nasci

SALVATORE J. NASCI
Project Engineer

APPROVED: Fred I. Diamond

FRED I. DIAMOND
Technical Director
Communications and Control Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (DCLD), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-79-278	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): 16 KB/S DATA MODEM PARTITIONING	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, 26 Aug 77 — 20 Jan 78
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Fred C. Killmeyer, Daniel D. McRae	8. CONTRACT OR GRANT NUMBER(s): F30602-77-C-0190
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Harris Corporation, Electronic Systems Division, Box 37, Melbourne FL 32901	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 33126F, 21570107
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (DCLD), Griffiss AFB NY 13441	12. REPORT DATE: November 1979
	13. NUMBER OF PAGES: 54
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same	15. SECURITY CLASS (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Distribution limited to U.S. Government agencies only; test and evaluation; November 1979. Other requests for this document must be referred to RADC (DCLD), Griffiss AFB NY 13441.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Salvatore J. Nasci (DCLD)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Modem	8 kb/s Modem
Wireline	Microprocessor
Secure Voice	LSI
Adaptive Equalizer	Partitioning
16 kb/s Model	HF Tests

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The primary objective of this program was to study techniques to reduce the size and cost of the 16 kb/s modem. These techniques involved custom LSI and/or standard microprocessor implementation of the various functions involved in the 16 kb/s modem.

On previous contracts a modem for 16 kb/s operation over the AUTOVON network was developed (Contract F30602-75-C-0129) and was extensively tested over the OCONUS network (Contract F30602-76-C-0460). Results of both (Cont'd)

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 112

Item 20 (Cont'd)

these efforts were very encouraging, and a contract (Contract F30602-77-C-0190) was awarded to investigate cost-effective means for future production efforts.

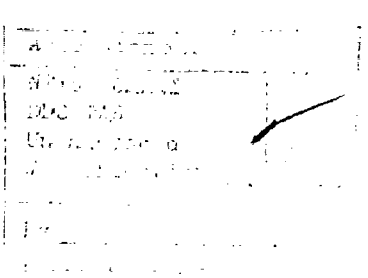
Results of this investigation have shown that a cost/size/parts reduction of 33 to 50 percent would be possible with the application of commercially available bit slice microprocessor chips and some custom LSI devices. Also the use of a processor for signal processing results in a flexible solution that can adapt to changing requirements and track the latest state of the art techniques with minimum hardware changes.

Additionally, by incorporating digital filtering rather than analog filtering, critical parameters can be better controlled for improved performance.

Thus, implementation with a microprocessor appears to result in both a more cost-effective approach as well as a higher performance solution of the 16 kb/s modem design. It appears prudent to undertake an additional effort to construct a feasibility model to confirm the findings of this report.

An additional phase of this contract involved testing of the 16 kb/s modem over simulated and actual HF links. Results of these tests were discouraging and indicated that in general the present design is not adequate for operation over these links. However, the testing did provide valuable data relative to revised design parameters that would result in a 16 kb HF modem.

TABLE OF CONTENTS



LIST OF ILLUSTRATIONS

LIST OF TABLES

EVALUATION

The Department of Defense requirement for an improved, inexpensive, end-to-end secure voice communications system using the existing analog AUTOVON Network has fostered the investigation of various secure voice transmission approaches. One such approach being considered is the 16 KBPS Continuously Variable-Slope Delta Modulation (CVSD) Digital Voice Technique which offers much improved speaker recognition and voice intelligibility as compared to that obtained with slower data rate digital voice systems presently available. This technique and moreover the use of existing narrowband (4 kHz) voice channels of the AUTOVON Network have created the need for a 16 KB/S Modem.

Under a previous CY-75 program, practical 16 KBPS Modem techniques were developed to provide satisfactory voice quality with 16 KBPS CVSD on a large percentage of dialed-up calls on the AUTOVON Network. The 16 Kilobit Modem Evaluation Program was responsible for providing the first in-depth step in determining the adequacy of the 16 KBPS Modem design. Reliable modem operation was obtained over AUTOVON facilities in the CONUS, Europe and the Pacific, as well as over Transoceanic (satellite and subcable) transmission links.

This report covers the implementation of the state-of-the-art microprocessor technology in the successful 16 KB/S Modem design. Additionally, testing of the modem over an actual HF link is a part of this report and it provides an insight as to the design parameter revisions that could result in a successful HF 16 KB/S Modem.

As a result of this Partitioning Study a Low-Rate Initial Production (LRIP) is underway under Contract F30602-78-C-0273 for one hundred (100) Modem Digital Data, AN/GSC-38.

Salvatore J. Nasci

SALVATORE J. NASCI
Project Engineer

1.0 INTRODUCTION AND BACKGROUND

1.1 Prior Study Activity

In February 1975 Harris Corporation was awarded a Contract (F30602-75-C-0129) from the Rome Air Development Center (RADC) to study approaches available for developing a 16 kb/s Modem capable of providing adequate voice quality over the worldwide AUTOVON network when operated with 16 kb/s continuously variable slope delta modulators (CVSD). The principal outcome of that study activity was a breadboard modem which was tested on a number of CONUS AUTOVON circuits from RADC with encouraging results. The results of that study, as well as the test results, are provided in the study final report.[1]

A subsequent Contract (F30602-76-C-0460) was awarded to the Harris Corporation by RADC in November 1976 to optimize certain of the modem algorithms and to peform extensive testing of the modem in the OCONUS network. The results of the second study, like the first, were very encouraging and provided a high level of confidence in the modem's ability to provide good voice quality over the existing AUTOVON network, as well as over future network improvements. The results of that study, as well as the test results, are provided in the study final report.[2]

1.2 Objectives

The two previous contracts had encouraging results and provided the basis for a third 16 kb/s Modem Contract (F30602-77-C-0100) from RADC with the following objectives.

a. Advanced Solid-State Study
   Peform a preliminary partitioning study of the baseline modem design with the overall goal being recommendations of design improvements/modifications resulting in cost-effective approaches for additional consideration in future production contracts.

b. HF Test

Test the 16 kb/s Modem on a HF simulator and over actual HF Links.

c. Modem Equipments

Provide additional equipments for use in planned OCONUS Biplexer Tests.

d. Biplexer Test Support

Provide support at OCONUS Biplexer tests.

Of the above objectives, only a. and b. will be discussed in this report.

1.3 Report Organization

Section 2.0 discusses the approaches taken and the results, in detail, of the Advanced Solid-State Study. The HF Test activity is discussed in Section 3.0. Conclusions and recommendations are provided in Section 4.0.

2.0 ADVANCED SOLID-STATE STUDY

2.1 Summary

The objective of the Advanced Solid-State Study was to examine the design and organization of the present 16 kb/s Modem and establish feasible alternatives to reducing its size and cost. These alternatives would then be investigated in more detail in a future program.

The objectives were met and a feasible approach was determined using available Advanced Micro Devices 2901 bit-slice microprocessors in place of the existing discrete processor. This approach appears to offer an overall recurring cost/complexity/size reduction of approximately 33 to 50 percent. Implementation of this approach to achieve the above reductions requires generation of new software as well as possibly development of some Custom LSI devices. It is not possible, as a result of this study effort, to accurately estimate the development cost, however, it does appear attractive enough to justify additional investigation in a future contract.

2.2 Approach

The overall goal of the Advanced Solid-State Study was to establish a basis for determining the most cost-effective means of implementing future engineering and production models of the 16 kb/s Modem. In order to determine this, it was necessary to first partition the modem structure in terms of circuit functions and assign some "relative cost" to each function. A block diagram of the transmit portion of the present 16 kb/s Modem is shown in Figure 1 and the receive portion in Figure 2. Cost was assigned as an integrated circuit package count for digital functions and actual dollar cost (per 1000 units) for known high dollar cost items such as the low-pass filter modules and the stable crystal oscillator. Integrated circuit count might not sound like a valid cost estimate since some integrated circuits cost considerably more than others, but functions that required a large quantity of digital integrated circuits also used the more costly packages as a result of previous

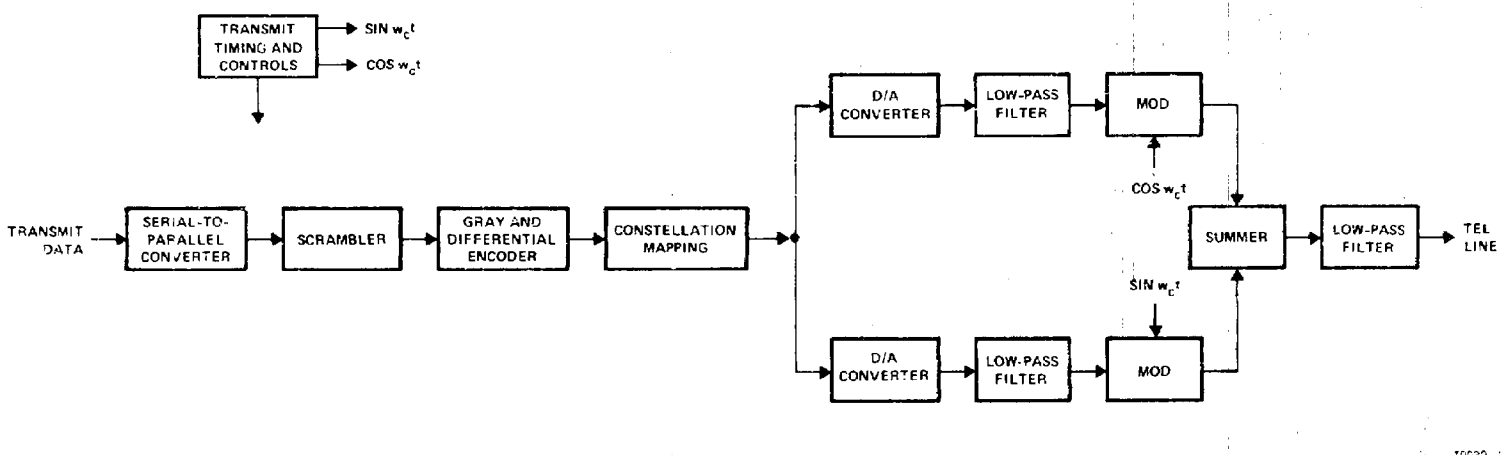


Figure 1. 16 kb/s Modem Transmitter Block Diagram

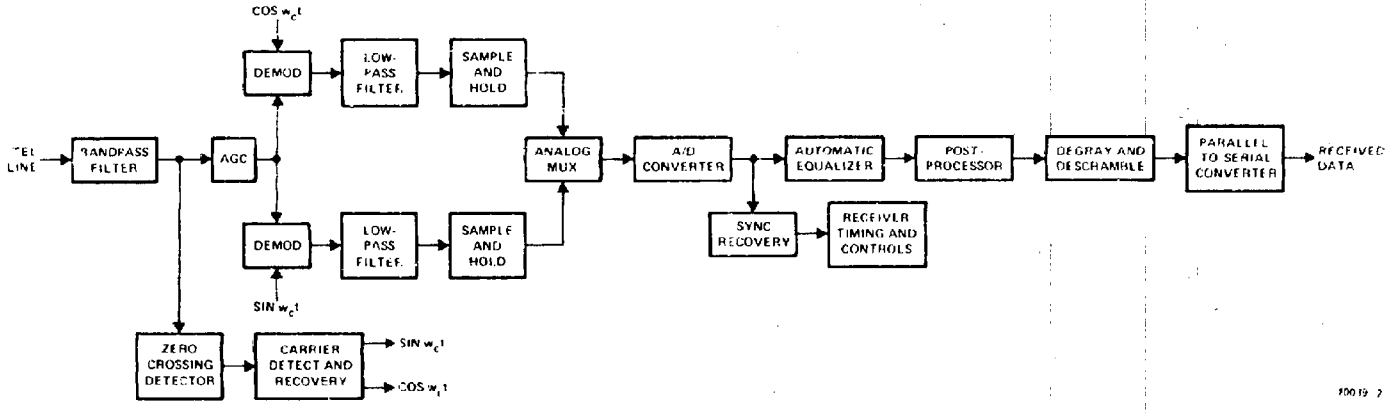


Figure 2. 16 kb/s Modem Receiver Block Diagram

c i . in reducing parts count and cost. Thus, functions that required mar ntegrated circuits (the automatic equalizer and post processor) also used the more expensive packages making it even more desirable to concentrate on these functions. Table 1 shows the various major functions in order of relative cost. This table was obtained by first grouping functions from the block diagrams for the transmit and receive portion of the modem; then the relative cost was assigned by a detailed review of the schematics to determine high-cost components and arrive at package counts for various functions. The total digital IC count for the modem is 356 with 140 IC's used in the post processor and 93 IC's used in the automatic equalizer; for a total of 233 IC's to accomplish these two functions and 123 IC's for the rest of the modem functions.

Table 1. Relative Cost of Various Modem Functions

1. Post Processing (140 IC's)
2. Automatic Equalizer (93 IC's)
3. Stable Oscillator ($112)*
4. Transmit and Receive Low-Pass Filters ($108)*
5. Carrier Recovery Loop (55 IC's)
6. Controls and Timing (25 IC's)
7. Scrambler/Descrambler, Gray Encoder/Decoder, Differential Encoder/Decoder (23 IC's)
8. Input/Output (minimal)
9 Modulator/Demodulator (minimal)

* Pricing based on 1000 units

In general, the approach in this study was to allocate study effort in proportion to relative cost as shown in Table 1. Consistent with this the functions will be discussed in more detail in the same order.

2.3 Post Processing

The Post Processing section consists of a special purpose processor that executes several fixed algorithms. These algorithms allow

the processor to generate outputs for equalizer coefficient updating, carrier-phase tracking, and received data. The equalizer coefficient updating output is used by the automatic equalizer to adjust the equalizer coefficients (or weights) when the modem is in the data transmission mode. The carrier-phase tracking output is used as a phase error signal by the carrier-tracking loop when the modem is in the data mode. The received data output is further decoded to generate the actual received data from the modem.

These algorithms are contained in the main program as a set of instructions that are executed sequentially. Along with instructions the program contains memory address locations containing data or constants. Instructions are further decoded by the processor to generate microinstructions which actually control the processor hardware necessary to implement a given operation. Microinstructions are generally referred to as the microcode and for each instruction there is a fixed set of microcode used to implement that instruction. Another way to look at this type processor is that instructions (ADD, READ, JUMP, etc) that are understood by the programmer have to be executed in the hardware as a series of basic data operations, similar to a subroutine, where the subroutines are in a different language than the main program. For example the operation X + Y = Z would be accomplished as follows:

1. Main Program

   a. ADD
   b. X
   c. Y
   d. Z

   } All 8-bit codes

2. Microcode

   a. Load operation code ADD to get microcode for this instruction,
   b. put value of X in RAM address,
   c. get data in RAM address X and put in register A,
   d. put value of Y in RAM address,

e. get data in RAM address Y and put in register B,
f. add register A to register B and put result in register C,
g. put value of Z in RAM address,
h. get data from register C and write in RAM address Z,
i. ready for next instruction from main program.

Subparagraph 2 is a simplification of what actually happens but it shows all the operations that have to be executed to perform one instruction. Each step of microcode takes one or more microcycles and the ADD instruction ends up requiring 13 microcycles. Since instructions are written so they can be used in many different programs, their execution has to be general in nature, i.e., even if the value of X was the output from the previous instruction it still must be written into RAM at the end of that instruction and read back out at the start of this instruction. Unfortunately, many times this is the case with signal processing, which tends to take some given data and manipulate it to get the desired output, similar to an analog signal passing through the transmitter or receiver. Because of this, for many algorithms it would be advantageous to put the output data into register A or B; rather than into register C, then RAM and back into A or B. If this were possible it would eliminate writing the data into RAM and reading it back out for the next instruction. Since addressing RAM and reading and/or writing to RAM requires most of the instruction time it would be advantageous to eliminate these operations as often as possible.

2.3.1 Present Post Processor Implementation

The Post Processor is presently implemented with a combination of MSI/SSI integrated circuits along with some LSI PROM's and RAM's. As mentioned earlier this requires approximately 140 digital packages and would seem like an appropriate area in which to use LSI techniques to reduce parts count, size, and power while increasing reliability and maintainability. Since it has a fairly standard architecture this is an area where a standard commercial microprocessor solution might be possible. The first solution that is suggested is replacing the present processor with one of the standard MOS microprocessors of the Intel 8080, Motorola 6800 or Zilog Z80 types. With these processors the basic limitation is speed. Not only do they have fixed instruction sets like the present processor but also their instruction sets are of a more general nature. This is inherent in that they are designed to handle a wide range of general purpose functions, whereas, the present processor has an instruction set tailored to its use in the modem. Also the general purpose processors are all using MOS technology which is not as fast as the bipolar chips in the present processor.

All this points to a bipolar bit slice processor such as the AMD 2901 with microprogrammed software. This device offers the speed of bipolar hardware and the flexibility of microinstruction programming. Also by cascading 2901 processors, the effective processor width can be either 8 or 12 bits depending upon system requirements. To give a better understanding of what the relative speeds of the different approaches are, Table 2 shows the time to add two numbers from memory using various processors. The table is only provided to show the basic speed of the processors for a given general instruction. In calculating the relative speeds, the two numbers to be added were loaded from memory, added, and then stored back into memory. This is the required procedure for adding in the present processor, however with the 2901 and its 16 internal registers, the register-to-register add time is only 0.2 microsecond. Under these conditions an improvement of over 6 to 1 in speed is possible. Couple this with the fact that each microinstruction is basically customized for the function being performed and the improvement margin increases.

Table 2. ADD Times for Various Processor Implementations

Processor	Technology	Bits	Time (μsec)
Intel 8080A-1	MOS	8	6.72
Motorola	MOS	8	10
Zilog Z80A	MOS	8	5.25
TI TMS 9900	MOS	16	7.8
TI SBP 9900	I^2L	16	13
Present 16 kb/s Processor	Bipolar STTL	8	1.3
AMD 2901 Processor	Bipolar LSTTL	8 or 12	0.8

2.3.2 Bit Slice Processor Selection

Having narrowed our approach down to the use of a bipolar bit slice proccessor, rather than the MOS and I^2L type microprocessors, the specific bit slice chip must now be selected. Since its introduction in 1975 the Advanced Micro Devices 2901 has rapidly become the industry standard for bit slice processors. This can be clearly seen by the large number of second sources that have been developed and the many articles written discussing its use by various Government agencies and industries. Because of this and its capability to satisfy the 16 kb/s modem processor requirements it would be the logical choice. It must be pointed out though that a new bit slice, the 2903, has recently been introduced by Advanced Micro Devices. This unit is similar to the 2901 but with additional capabilities which could further reduce the chip count of the processor. However this device is new and would require additional investigation, also no second source has yet been announced. Another possibility would be the Motorola 10800 bit slice. This is an ECL bit slice with a second source, however, ECL is harder to work with than low-power Schottky TTL (2901), fewer support chips are available and the parts are more costly; but it is capable of running at higher speeds so more work could be done in the processor. For the following discussion the 2901 will be used since it appears the most likely candidate although all three chips are very similar in their structure and uses.

2.3.3 Bit Slice Processor Description

Figure 3 shows the block diagram of the most feasible microprocessor approach for the 16 kb/s Modem. The control PROM section serves primarily as a place to store microinstructions; the coded pieces of data that direct the activities of the other blocks in the processor. The group of logically related microinstructions stored in the memory is referred to as the program. The controller sequences the control PROM through a logical sequence to process the program. Data from the control PROM output in turn tells the controller what address to use for the next microinstruction. The test condition multiplexer is used to do conditional jumps where the instruction directs going to a branch address location if the condition is true or the next location if the condition is false.

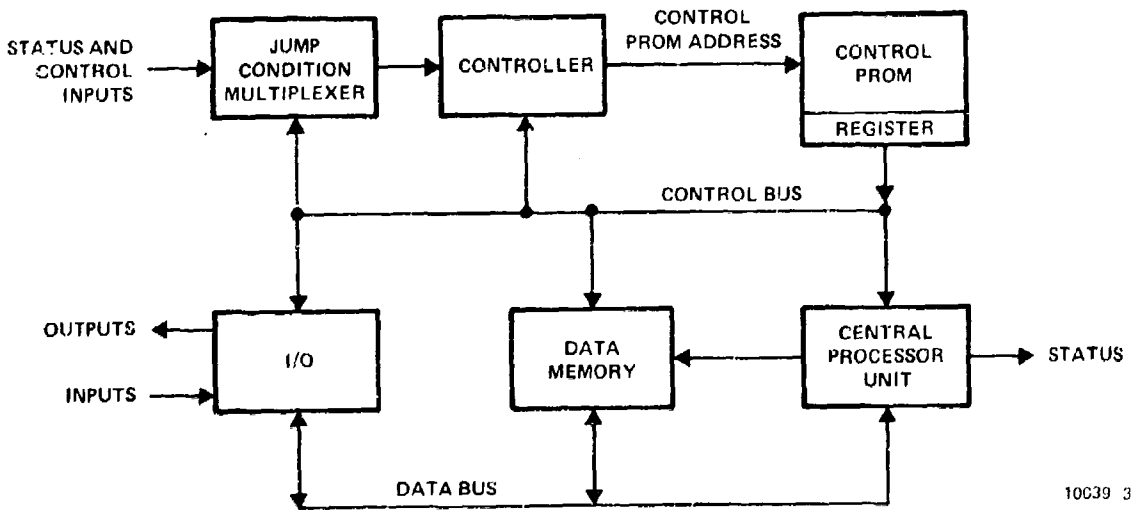


Figure 3. 16 kb/s Modem Processor Architecture

The heart of the processor is the Central Processor Unit (CPU). This contains the 2901 four-bit slice chips along with additional chips for carry look ahead, fast addition, and some multiplexer IC's for routing of double length shifts and software multiply control. The block diagram of a 2901 four-bit bipolar microprocessor slice is shown in Figure 4. This is an LSI chip with all the basic CPU functions including: a four-bit arithmetic/logic unit, 16 general purpose four-bit registers, a four-bit-Q (extension) register, and multiplexer circuitry to implement the RAM and Q shifts and the ALU data source selection.

The general purpose register file is designed with a latched multiport input structure that permits reading from two RAM locations simultaneously, operating on them in the ALU, and either storing the results (or the results shifted left or right one bit) back in the RAM and/or outputting the results at the three-state output (Y) - all in a single synchronous clock cycle. Control commands are entered on the slice through an encoded nine-bit microinstruction field, which controls operand and function selection for the Register Arithmetic Logic Unit (RALU) and provides destination and shift operations for controlling internal registers. The four-bit (D) input port is for data entry, while the three-state (Y) output port connects the RALU output to an external system bus.

2.3.4 Post Processor Results

With the use of the 2901 bit slice microprocessor the integrated circuit package count of the processor is reduced from 140 to 60. Also the bit slice processor is capable of solving the present algorithms in approximately half the time of the present processor. This means that there is time left to handle other functions that are not presently included in the processor. The only price for these functions would be that of larger PROM's and of developing the software to implement the various functions. An advantage to this is that these additional functions would be under software control which would make modifications relatively easy since the programs could be changed by altering the information stored in the PROM's.

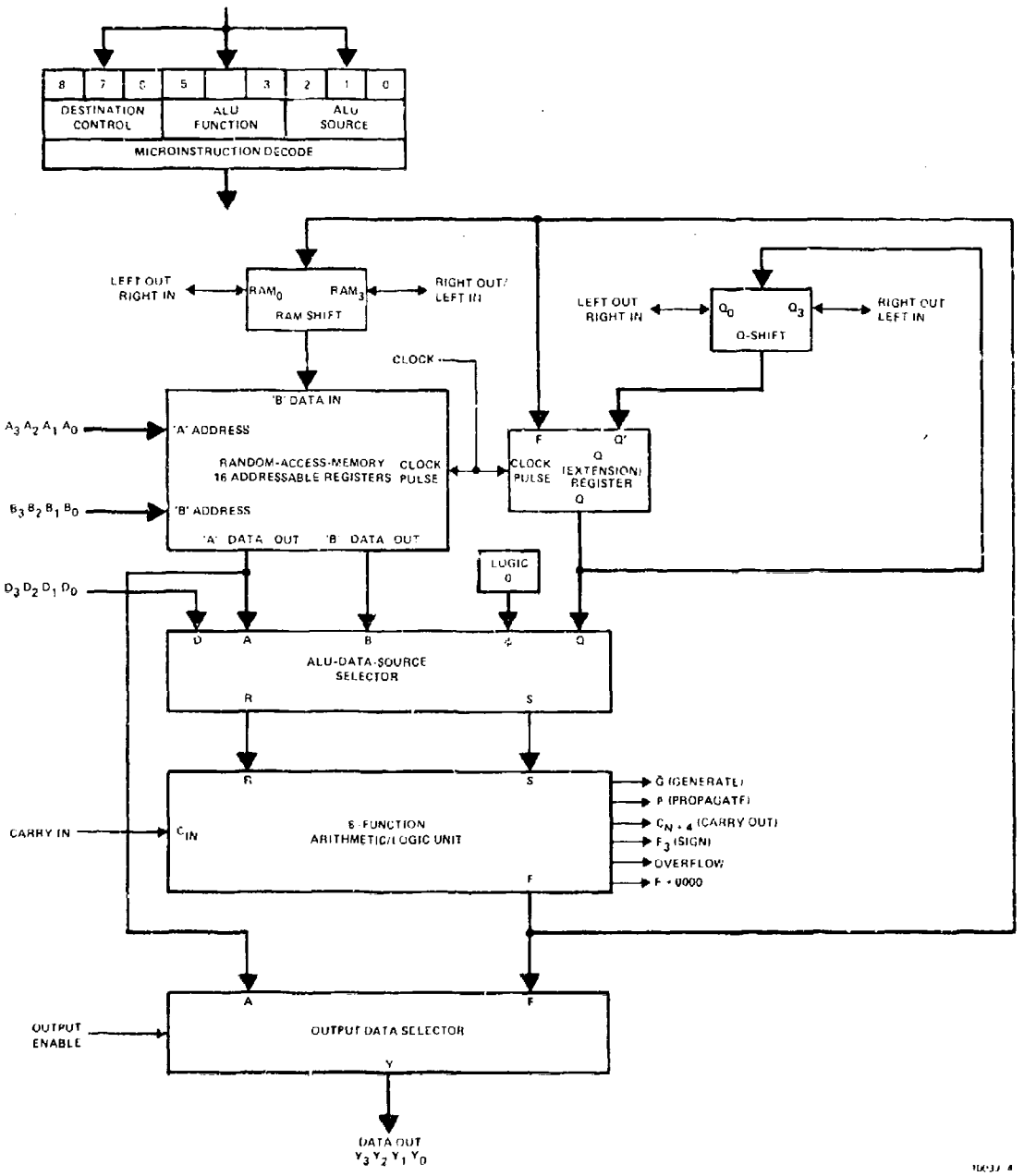


Figure 4. 2901 Bipolar Microprocessor Slice Block Diagram

2.4 Low-Pass Filters

2.4.1 Present Implementation

The 16 kb/s modem presently has four critical filters that are six-pole Butterworth analog low-pass filters with a cutoff frequency of 1400 Hz. Two are used before the modulator in the transmitter and two more are used after the demodulator in the receiver. The basic requirements of these filters are sharp cutoff and linear phase. The sharp cutoff is required to stop the harmonics of the data from being modulated in the transmitter and reduce the harmonics generated by the demodulator in the receiver. Linear phase is required to control the intersymbol interference. In the present design these requirements are relaxed to keep the cost within reason, relying on the automatic equalizer to adjust itself to minimize their effects. However the fact that the automatic equalizer has to compensate for the nonperfect filters uses up part of its range and therefore reduces its ability to handle badly distorted telephone lines. In addition, because the filters are analog they have the normal drift and stability problems associated with analog implementation.

2.4.2 Digital Filters

Digital filters have become more attractive with the decrease in cost of digital circuitry and the higher operating speeds of digital processors. These techniques allow the exact control of critical parameters with exceptional stability; properties not available with analog filters as discussed previously. Figure 5 shows the normal representation of the digital equivalent of an analog filter. From this block diagram it can be seen that the analog signals are converted to their digital representation, filtered in the digital processor, converted back to analog signals and then passed through a low-pass filter to remove the sampling frequency components. The crucial choices that have to be made in the design of the digital signal processor are the choice of the sampling rate, that is, the basic rate of processing of signals, and the

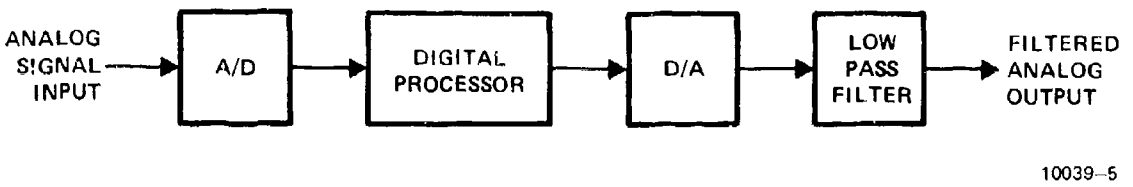


Figure 5. Digital Filter Block Diagram

number of bits of accuracy required during sampling, processing and at the D/A output. For the 16 kb/s modem analog signals, the highest frequencies of significance are about 3.5 kHz. Therefore, according to the Nyquist sampling theorem, a sampling rate of 7 kHz is enough to characterize and process all telephone line signals. However such a low sampling rate presents some drawbacks. Nonlinear operations have to be considered carefully because of the risks of aliasing. Many operations are difficult if not impossible to perform at this low sampling rate. Also, low sampling rates impose very stringent requirements on the low-pass analog filter at the output of the D/A converter.

On the other hand, a low sampling rate allows the most efficient use of the digital signal processor since the larger the interval of time between samples, the more computations that may be performed for each sample. The modem low-pass filters occur near the transformation from digital-to-analog signals in the transmitter and from analog-to-digital signals in the receiver. Because of this if the digital processing is performed properly the A/D converter shown in Figure 8 can be eliminated in the transmitter, and the D/A converter eliminated in the receiver processing. At these interfaces the digital signals are being processed at the baud rate in the transmitter (2667 Hz) and twice the baud rate in the receiver (5333 Hz). With that in mind it can be seen that to keep the implementation simple, it is desirable to sample at a multiple of the baud rate in the transmitter and a multiple of twice the baud rate in the receiver. Also, it was shown previously that the sampling rate had to be at least 7 kHz. Since twice the baud rate is 5333.33 Hz the next

multiple of that would be 10666.67 Hz which was selected for the sampling rate for the digital low-pass filter design. This rate is also used for the transmit filter section to keep both implementations as similar as possible.

Having selected the sampling rate, the type of digital filter structure must now be determined. Digital filters fall in two broad classes, those whose current output depends only on a weighted sum of previous inputs and those whose current output depends not only on a weighted sum of all previous inputs, but also a weighted sum of all past outputs. The first type is referred to as a nonrecursive, transversal or finite-duration impulse response (FIR) filter while the second type is a recursive or infinite-duration impulse response (IIR) filter. The former filter has a transfer function of the form

$$H(Z) = \sum_{k=0}^{N-1} a_k Z^{-k}$$

and may be implemented in a nonrecursive realization where all the poles are located at Z = 0. The latter class of filters has a transfer function of the form

$$H(Z) = \frac{\sum_{k=0}^{M-1} b_k Z^{-k}}{\sum_{k=0}^{N-1} a_k Z^{-k}}$$

where the pole placement is arbitrary and must be implemented in a recursive realization. Nonrecursive filters are desirable since they can yield exactly linear phase characteristics. This is extremely important in high bit rate modems since phase distortion generates intersymbol interference that must be corrected by the automatic equalizer. Since the automatic equalizer has a limited range over which it can correct phase and amplitude distortion, the more linear the phase of the modem filters the more range left in the automatic equalizer to correct distortion in

the telephone lines. Because of this, nonrecursive filters are optimum for high-speed modem designs.

The nonrecursive filter has a representation of the form

$$y(kT) = \sum_{m=0}^{N-1} h(mT)x(kT - mT) \quad ; \quad k = 0,1,2,\ldots$$

where: $y(kT)$ is the output at kT

$x(kT - mT)$ is the input at $kT - mT$

$h(mT)$ is a sample of the analog filter inpulse response $h(t)$

T is the sampling interval $(1/f_s)$ and the results are evaluated at time kT

Expanding the above expression, it is easily seen that

$$y(kT) = h(0)x(kT) + h(T)x(kT-T) + \ldots + h\left[(N-1)T\right] x(kT-NT+T)$$

That is, the current output is obtained by taking the current input and multiplying it by $h(0)$ and adding that result to the previous input multiplied by $h(T)$, etc. In other words, the output response is just a weighted sum of present and past inputs. Pictorially, these equations are represen :d by Figure 6.

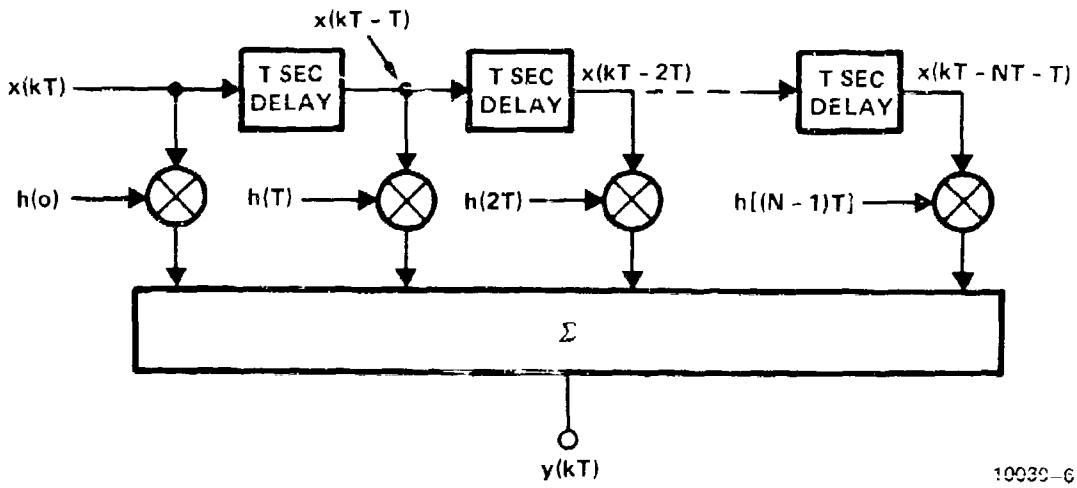


Figure 6. Nonrecursive Digital Filter Block Diagram

The ideal low-pass filter with cutoff frequency f_c has the transfer function

$$H(f) = \begin{cases} 1 \text{ for } |f| < f_c \\ 0 \text{ for } f_c < |f| < f_s/2 \end{cases}$$

Where f_s is the sampling frequency. To find the coefficients (h(0), h(T), . . .) for the nonrecursive filter we calculate the Fourier coefficients of H(f) which is done as follows:

$$h(NT) = \frac{1}{fs} \int_{-f_s/2}^{f_s/2} H(f)\, e^{j2\pi fnt}\, df$$

$$= \begin{cases} 2\, f_c/f_s & \text{for } N = 0 \\ \dfrac{\text{Sin } (2\pi\, Nf_c\, /f_s)}{N\pi} & \text{for } N \neq 0 \end{cases}$$

However it can be seen that there are an infinite number of values of h(nT) needed to represent H(f). Since we are restricted to a fixed number of coefficients; the smaller the better from the processing aspects, the coefficients must be truncated or limited to some minimum number while still doing the required filtering. To accomplish this the coefficient values are multiplied by a "window" function. This is the same as convolving the desired frequency response with the Fourier transform of the window. Thus the frequency response will be a "smeared" version of the desired response. To keep the required computations in the processor to a minimum the narrowest window possible is desired, but as the window is narrowed, the overall response begins to take on more of the characteristics of the window. This is especially true for a rectangular window and relates to Gibbs phenomenon. Therefore, a less abrupt truncation of the Fourier series is desired and can be accomplished by tapering the window smoothly to zero at each end. There are trade-offs involving the shape of the window function, number of coefficients required, transition width of the filter response, and stopband attenuation. An investigation of various window functions showed that the

Hamming Window gave the best stopband attenuation (> 40 dB) with the minimum transition width (16 fc/N, where N is the number of taps).

With the above conclusions a program was written to calculate the coefficients for various length delay lines, round off or truncate these values to various bit length accuracies and calculate and plot the frequency response for the parameters given. The sampling frequency was 10,667 Hz and the cutoff frequency 1333 Hz. By comparing the resultant plots with the six-pole Butterworth filter response and the modem requirements it was found that a 17 tap filter rounded off to eight-bits of accuracy would be better than the existing filters while not loading the processor heavily. The response of the new and existing filters are shown in Figure 7. It can be seen that both filters are the same through the passband area, the digital filter has a sharper transition band, and the analog filter has better stopband attenuation. However, a stopband attenuation of 40 dB is sufficient for our application, giving the digital filter an overall advantage since its transition band is narrower. Also the digital filter has linear phase; a condition not found in the analog filter.

2.4.2.1 Transmit Low-Pass Filters

Once the characteristics of the digital low-pass filters had been arrived at it was then time to look at the actual implementation of the filters in the modem. Although the transmit filters are required to operate at a sampling rate of 10,667 Hz, the input data to the filters is changing at the baud rate (2667 Hz). This means that within the digital filter the same input sample occupies four adjacent storage words in the delay line. Therefore four coefficients are multiplied by the same sample value. To reduce the number of multipliers required we could sum the four coefficients together before multiplying them by the sample value. We have now reduced the number of multiples required to 25 percent of the straightforward approach but the number of additions has increased. Since the processor can do addition faster than multiplication this is a good trade. These additions can also be eliminated if rather than store all the coefficients and add groups of four together, we add groups of

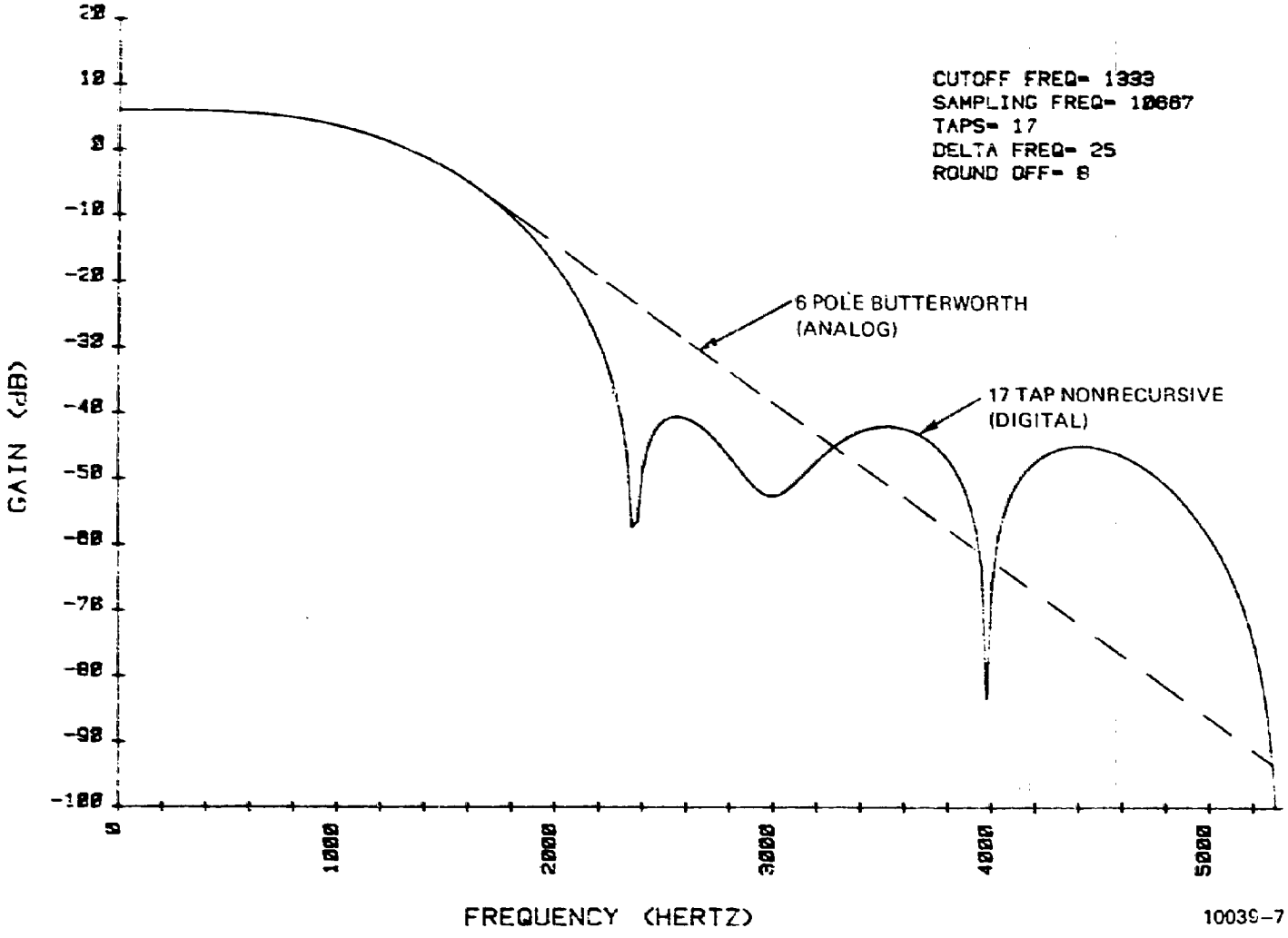


Figure 7. Low Pass Filter Response

four together and store them as the coefficients for the filter. Since the four coefficients to be added change every sample period (but repeat after four samples) we can store four different sets of coefficients representing the four different groupings of coefficients for each sample. Before going on to calculate the final operations of the filter we should point out that the 17 tap digital low-pass filter has zero for the two end coefficients, therefore only 15 coefficients are required to generate a 17 tap filter. Because of this and the fact that we can group four coefficients together, there are only five multiplies and five adds required to generate two of the filtered samples and four multiplies and four adds required to generate the other two filtered samples.

Another fact that can be used is the limited values of the input data. Since it is generated by the mapping for the four amplitude 16 phase constallation it has to be one of 64 different values. However since there are two channels in the transmitter where the filters are used there are only 32 possible values for each filter input. Half of these are positive and the other half negative yielding ± 16 values. Thus all the data inputs could be represented by four bits plus sign.

To eliminate all multiplies in the processor we could store the product of all possible data inputs multiplied by all possible filter coefficient groupings in a PROM and use the data combined with the coefficient group to address the PROM. The number of possible values for coefficient groups is eight. This is lower than might be expected due to the fact that the coefficient values are symmetrical about the center of the delay line and that every fourth coefficient away from the center coefficient is zero. Therefore the address of the coefficient value can be represented by three bits and the data by four bits plus sign for a total of seven bits plus sign. This results in a 128 x 8 bit PROM to store all the products to eight bits of accuracy and the sign bit is used to determine whether to add or subtract the PROM output from the sample accumulator. The time required by the processor to handle the transmit filter operations is approximately 35 microseconds or 9.5 percent of the available processing time. Remembering that the postprocessing took 50

percent of the time we have now used up 59.5 percent of the processing capability of the processor.

2.4.2.2 Receive Low-Pass Filters

We have demonstrated logical approaches to reduce processor time in the implementation of the transmit filters, now let's look at the receive filters. First we note that the equalizer processes data at 5333 Hz so only half the filter outputs are necessary since the filter is generating outputs at a 10,667 Hz rate. Also the data input to the filter is 7 bits plus sign and there are eight different values (three bits) for the coefficients. Thus to look up the products in a PROM requires 1K x 8 of storage. This is considerably more than for the transmit filter but the only alternative is a hardware multiplier. Since 1K x 8 of PROM is cheaper than a 8 x 8 hardware multiplier we will option for the PROM. Because the data sample inputs are independent we cannot group coefficients as in the transmitter. Based on the need to look up products in the table for each coefficient the receive low-pass filter operation will take approximately 70 microseconds or 18.5 percent of the processor time. Adding this to the time for postprocessing and the transmit filter functions we have now used up 78 percent of the processor's capability. This indicates there is the possibility of doing additional functions in the processor so other areas should be studied.

Before going on to the Automatic Equalizer it is desirable to see if the modulator and demodulator functions could be handled in the processor. This is required for the demodulator section if we are to simplify the Automatic Equalizer and therefore it has to be solved before proceeding. Also since the modulator is basically the same as the demodulator they will both be treated as one function as far as the processor is concerned. In order to demodulate the received signal we must first decide the frequency and phase of the received carrier. That is the job of the carrier recovery and will be used as an input to the demodulator. For the modulator the carrier frequency and phase is generated by taking the previous carrier phase and adding to that the phase shift the carrier goes through between samples. This operation is

repeated at the sampling rate to yield transmit carrier information. Now that we have the carrier for both the modulator and demodulator it is necessary to do the multiplications necessary for modulation. In order to keep the harmonic content as low as possible we will look up the sine and cosine values of the carrier angle in a PROM at the sampling rate. For accuracy in the receiver, values of multiples of 0.75^{o} should be stored in the PROM from 0 to 90^{o}. This requires 512 x 8 bits of PROM and can be used for both modulation and demodulation. Since we are dealing with multiplication operations at the sampling rate this function requires approximately 30 microseconds for modulation and another 30 microseconds for demodulation or a total time of 60 microseconds in the processor. This is another 16 percent of the processor's available time and brings our total processor loading to 94 percent. It is debatable whether this is too much of a load from the standpoint of just studying the problem and not having implemented any of these routines. However we wished to proceed with our study of other areas, although there will have to be trade-offs made regarding which functions to implement in the processor and which to implement with hardware. Half the processor time is used for the postprocessing functions. This is a conservative estimate and may be as low as 25 percent, in which case additional functions could be put in the processor. Also in order to decide which functions to place in the processor we must know how much processor time they require. Therefore all functions that appear feasible for processor implementation will be studied and their placement decided when actual times are known.

2.5 Automatic Equalizer

The Automatic Equalizer is where the original work was done that spawned the 16 kb/s modem idea. It is because of the techniques used in the equalizer that 16 kb/s operation is possible over even severely distorted telephone lines. Since it has been an area of extensive investigation and redesign, the implementation already involves LSI chips in the form of the 12 x 12 multiplier and RAM storage. Also because of the very high processing rates involved and the length of the tapped delay line it doesn't lend itself to other LSI techniques except for the control section. Here either a custom LSI chip or possibly a simple processor could be used to control the functions of the automatic equalizer. Another solution would be to let the processor control the equalizer by extending the length of the microcode words. This might prove effective since the equalizer function involves repeating the same basic operation many times. It should be pointed out that with the postprocessing functions being performed in the bit slice processor, the automatic equalizer section of the modem is now the most costly function. Therefore this area should be looked at rather closely.

A block diagram of the present automatic equalizer is shown in Figure 8. From our previous discussion we know weight memory B is no longer necessary since the sine and cosine filter and demodulator channels are identical. This eliminates five 256 x 4 bit RAM's. The present multiplier in the automatic equalizer is the TRW MPY-12AJ 12-bit parallel multiplier. This is an LSI chip using a triple-diffusion bipolar process in an emitter-follower-logic configuration to yield high densities and high speed. However a new 12-bit parallel multiplier-accumulator chip the TRW TDC 1003J, does all the functions of the present multiplier plus addition or subtraction to an accumulator. This allows the functions labeled Accumulate Sum, Accumulate Register, and Error X Data Register to be performed by the new multiplier-accumulator. By doing this another 11 integrated circuits have been eliminated from the modem. To further reduce the integrated circuit count in the automatic equalizer the ten packages associated with the error signal from the processor could be replaced with two IC's and allow the processor to decide whether the X or

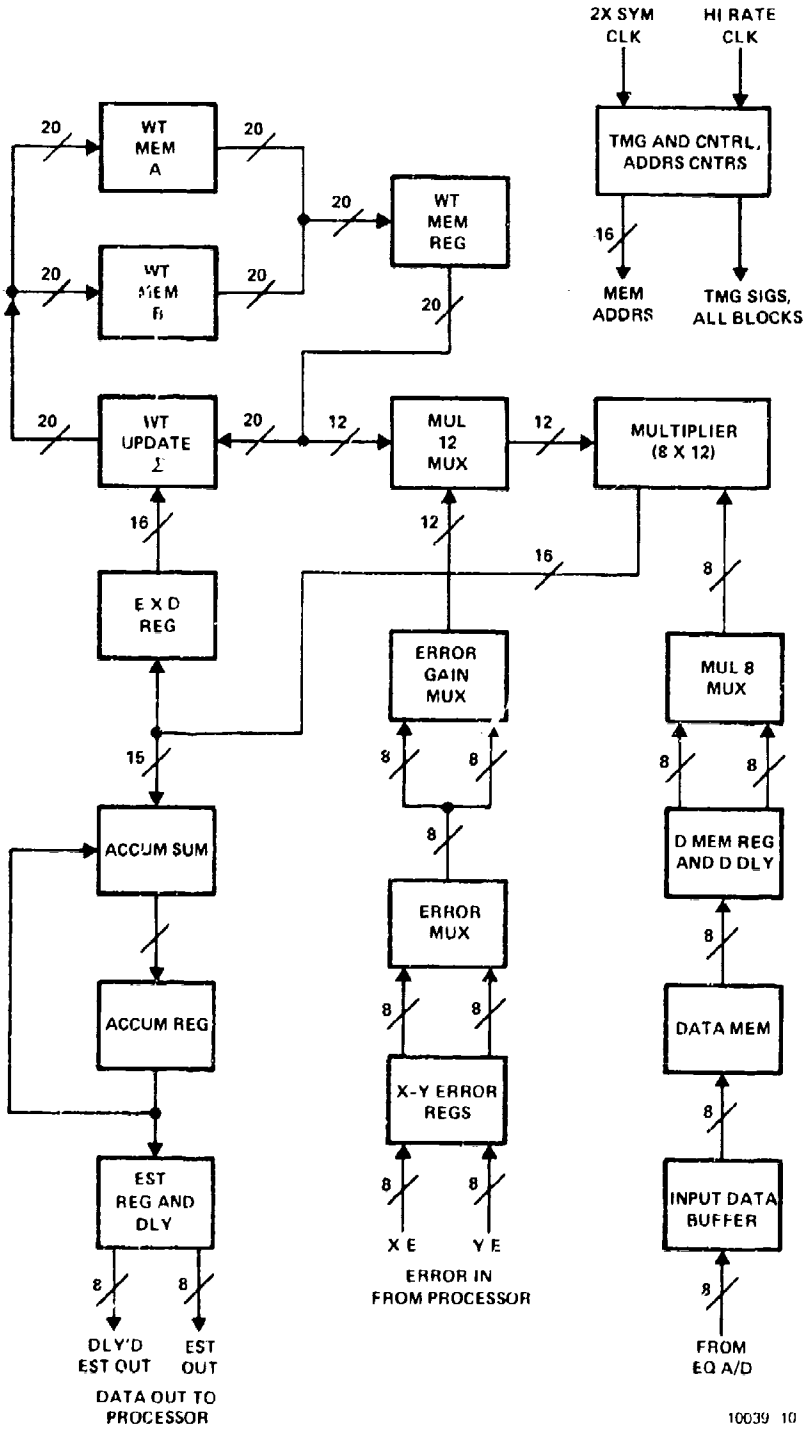


Figure 8. Automatic Equalizer Block Diagram

Y error signal was required and what value the error gain should have. Also since the input data will be coming from the processor the input data buffer in the automatic equalizer will no longer be needed. This reduces the IC count by another five packages. The estimate and delayed estimate registers could be eliminated, to save three IC's by putting this function in the processor. With the moving of some functions to the processor and the simplification of other areas the number of IC's required in the timing and controls area of the equalizer is reduced from 29 to 25. This is still an area for investigation and either a LSI sequencer or custom LSI solution should reduce this section even further. Some other minor savings might also be possible by going to the more efficient 20-pin packages and this could be done in any redesign with minimum effort. In conclusion, the equalizer does not lend itself to either of the standard "order of magnitude" reduction techniques, custom LSI or microprocessor implementation, except possibly in the timing and controls area. However, we have been able to reduce the chip count from 93 to 57 which should result in significant savings in cost and size.

Another possible solution would be to divide the automatic equalizer into modular sections similar to the technique described by Logan and Forney.[3] They selected eight tap sections for each module since that was the complexity that would fit in custom MOS/LSI. This would still appear to be a good number for our implementation since the advances in semiconductor technology should balance out with the requirement to process slightly more than twice as many samples/baud than they were processing. However, for a 256 tap equalizer 32 custom LSI chips would be required plus five standard packages for miscellaneous functions. This results in a total of 37 packages, most of which are custom LSI. If custom LSI were applied to the previously discussed equalizer timing and controls area then the chip count of 57 should reduce to 34 with only one custom LSI chip used. So this approach not only uses fewer chips but, on the average, the chips should be far less costly.

2.6 Stable Oscillator

The Stable Oscillator function is one area of the 16 kb/s modem that is rather expensive but doesn't lend itself to a cheaper solution. The local receive clock is not phase and frequency locked to the remote transmit clock. This is possible since the automatic equalizer is powerful enough to adjust for clock sampling error over a very wide range. However there are certain restrictions that must be met for proper operation. These involve the frequency stability of the two clock sources. If the clocks drift too fast in phase relative to one and the other the equalizer cannot adjust fast enough to track out this movement. Therefore the rate of change of phase has to be controlled. Also the long term frequency drift has to be low if the equalizer is to maintain its operation. This is caused by the fact that the automatic equalizer has to act as a buffer for the differences between transmit and receive clock. It does this by allowing the center tap to move up and/or down the delay line to track the differences in the two clocks. However if there is a constant frequency offset in one direction the center tap will eventually move to one end of the delay line and the equalizer will no longer be able to compensate for both clock offset and the other impairments it is correcting. With carrier systems where all clocks are phase locked to one another and the modems are locked to these clocks this is not a problem. However in systems where this is not the case the modem operating time for each call is a function of the accuracy of the oscillators it uses. Therefore to ensure long operating times for each call the oscillators have to be very stable. To accomplish this oven stabilized oscillators are required.

One possible solution would be to use a cheaper oscillator and buffer the received signal with a first in-first out (FIFO) memory. In order for this technique to operate properly we must regenerate the transmit clock to load data into the FIFO. Also this has a finite operating time before the FIFO overflows. To overcome this it might be better to use the regenerated transmit clock for the receive clock in a somewhat standard approach. Although this requires additional hardware the overall cost may be less than using the present stable oscillator. It

should be pointed out that this solution complicates the mixing of transmit and receive functions in the processor since they may not be synchronous at the same location.

2.7 Carrier Recovery Loop

This function is necessary to ensure that the same carrier frequency and phase is used for demodulation in the receiver that was used for modulation by the transmitter. It also functions as a frequency sensitive carrier detector. During the initial period of receiver start-up this section detects the presence of unmodulated carrier and then measures the frequency and phase of the received carrier. This information is used to set the divide value of a numerically controlled oscillator. After the initial acquisition this counter is fixed and all tracking is maintained by the post processor. Since there is no other processing being done in the receiver during the carrier acquisition period, all the processor time is available for this function. Because of this the carrier recovery section can be implemented in software in the processor leaving only the hardware for the zero crossing detector required. Since the demodulation will be performed in the processor the only information that is required is the carrier phase at each sample. This is obtained by measuring the initial carrier frequency and phase and every sample period adding the calculated phase shift modulo 360 degrees. Thus the 55 IC's presently required for this function should reduce to three to handle the zero crossing detector and the interface to the processor. The processor time is free since very little else would be happening in the processor during this period. However, it should be pointed out that the basic processor design allows for 1K lines of microcode. If the number of lines of microcode exceed 1K then the cost of more program memory must be taken into account. Even if this did happen it would still be cheaper to implement this function in the processor since approximately 52 IC's are being replaced.

2.8 Control and Timing

There are control and timing functions associated with both the transmitter and receiver. With the use of the bit slice processor many of the controls become part of the program. That is, where the subroutine to perform a particular function is located in the main program controls when an input is sensed or an output is generated. Also many of the controls are used to pass data from one function to another at the proper time. This also is accomplished by the main program in the processor. Other controls, such as Request-to-Send are sensed by the jump condition multiplexer in the processor control area. This would be handled by checking the status of the Request-to-Send input to the multiplexer on a periodic basis and jumping to one subroutine if true and a different subroutine if false. The processor is designed to sense 32 different inputs through this multiplexer. Therefore, although this hardware was considered part of the processor in general, it is required for the control and timing functions.

Three techniques for handling the timing functions with the processor based modem were investigated. The first is to write the program in such a way that all timing is derived from the processor system clock. This would involve generating an output or sensing an input based on where that operation was performed in the program. To do this all possible paths through the program have to take the correct amount of time. Although this is possible it makes the software job considerably harder since all path times must be checked. It also complicates software changes for the same reason.

The second technique is similar to the first except part of the timing is generated external to the processor and inputted to the processor by means of the jump condition multiplexer. All the timing can be derived from the sampling clock. Therefore if external timing was used to generate the sampling rate, the program could be slaved to this timing by use of the jump condition multiplexer and all critical timing functions derived during transitions of the external clock. This requires a little more hardware (approximately four IC's) than the first approach but makes the software considerable easier.

A third technique involves generating all the timing external to the processor and interrupting the processor whenever a time critical input or output signal must be serviced. This approach requires the most hardware (approximately 20 IC's) and care must be taken when writing software, since interruptions can occur at anytime in the program.

It was felt that the second technique would be the most desirable solution for the reasons cited above concerning the hardware and software complexity. With this approach the control and timing functions require four IC's instead of 25 for a savings of 21 IC's in this area.

2.9 Miscellaneous Functions

This section will discuss the remaining sections of the 16 kb/s modem and how their cost can be reduced in future models.

The first area of discussion is that involving the Scrambler/Descrambler, Gray Encoder/Decoder, and Differential Encoder/Decoder. These areas are very similar in nature with transmit functions basically reversed in the receiver. Because of this a solution to the transmit functions will also work for the receive functions to the extent of this investigation. Two techniques for handling the Scrambler function in the processor appear feasible. The first involves the addition of an exclusive-OR gate to the processor to help simplify the generation of the pseudorandom pattern. This allows the processor to generate the pattern by shifting with a conditional carry input depending on the output of the exclusive-OR gate. A second technique involves looking up in PROM the next output sequence based on the last sequence. The first technique takes more processor time but less hardware cost than the second. Thus, which method we use will be a function of the time available in the processor. In either case this function will only take four microseconds or less of the processor's time. That is four microseconds to scramble and an additional four to descramble the data.

The Gray Encoder/Decoder and Differential Encode/Decode functions also can be implemented in the processor with a minimum of time and hardware. These two functions will require slightly less than three microseconds of processor time and approximately 16 locations of PROM lookup table for the Gray coding portion. Therefore in this section there are 23 IC's which can be replaced by two or three with a minimal amount of the processor's time.

The Input/Output area is covered in the processor design except for the Mil drivers and receivers. These should take a maximum of six packages which would be the same regardless of how the modem is implemented.

Another area that has been mentioned before but not discussed in detail is that of the Modulator/Demodulator. This is an area that

because of the many multiplies required for each symbol places a higher load on the processor than one would think based on previous implementations. However with the savings in the automatic equalizer and the way the carrier is generated for these functions it is still highly desirable to keep these functions in the processor. The processor time required for each function is 30 microseconds. This could be improved considerably if the multiplier in the automatic equalizer could be used rather than doing multiples in software. Further investigation should be undertaken to study the sharing of the multiplier between the processor and the automatic equalizer but time would not allow this on the present study.

When all the improvements are implemented, the Power Supply load should decrease approximately 33 percent. The reason the power supply doesn't decrease in relation to the chip count has to do with the higher than normal power requirements of the processor and memory IC's. Also the overall size should decrease by approximately 33 percent.

Another benefit is in the area of maintainability. Since the modem will be heavily processor oriented, software routines can be programmed for self-diagnostics. This should aid in decreased repair times both in the field and at repair depots.

2.10 Summary of Savings With Microprocessor Implementation

We have discussed how relative cost savings would be achieved in various functions by the use of a bit slice microprocessor, large capacity PROM's and RAM's, and possibly custom LSI. The results will now be reviewed briefly:

a. Post Processor IC count reduced from 140 to 60. This function only requires approximately 50 percent of processor time available.

b. Automatic Equalizer IC count reduced from 93 to 57 with possible further reduction of 23 chips using custom LSI.

c. Transmit and Receive Low-Pass Filters with a present cost of $108 at 1000 unit pricing reduced to a cost of

approximately $20. This function requires approximately 28 percent of the processor's time. (There is an additional reduction of one D/A in the transmitter and one sample-and-hold and one analog multiplexer in the receiver.)

d. Carrier Recovery and Detector reduced from 55 to 3 IC's with use of processor during normal function downtime.

e. Control and Timing reduced from 25 to 4 IC's with very little use of the processor's time (< 1%).

f. Scrambler/Descrambler, Gray Encoder/Decoder, and Differential Encoder/Decoder reduced from 23 to 3 IC's and requiring only 2 percent of processor's time.

g. Modulator/Demodulator replaced four analog switches and associated circuitry by using 16 percent of processor's time. (This also reduces complexity of automatic equalizer and analog low-pass filter in transmitter.)

h. Power Supply load reduced by 33 percent.

i. Size reduced by 33 percent.

This results in an overall reduction of IC packages from approximately 356 to approximately 150 plus elimination of $108 worth of filter modules, one D/A converter in the transmitter, four analog switches, and other miscellaneous hardware. However, since the 150 IC's would in general cost more than the present IC's, the total savings would be from 33 to 50 percent. The block diagram of the new modem transmitter is shown in Figure 9 and the receiver in Figure 10.

Up to this point, the study has shown that a significant cost reduction is feasible by use of an AMD 2901 microprocessor. However, the development cost in terms of dollars, time and risk have not been addressed.

The total processor loading is estimated from 75-100 percent of the time available. This doesn't leave much room for margin or future changes. There are alternatives if the final processor load is too great.

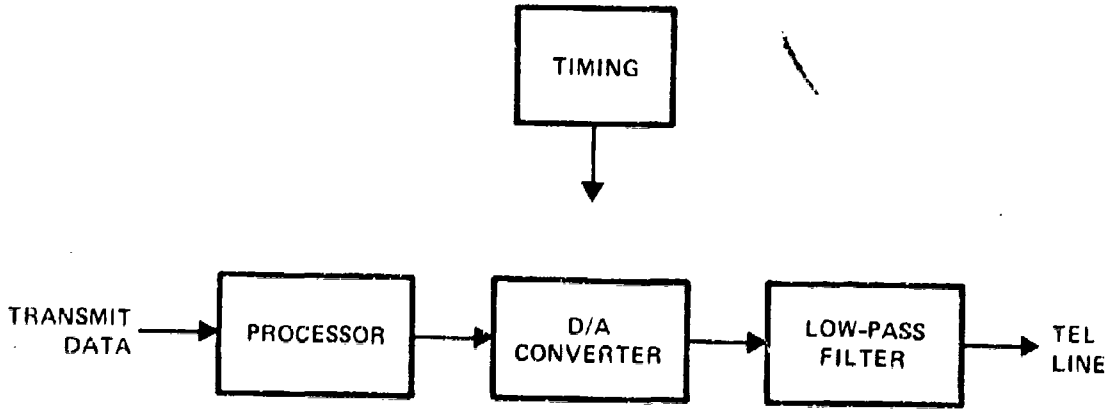


Figure 9. 16 kb/s Modem Transmitter With Processor

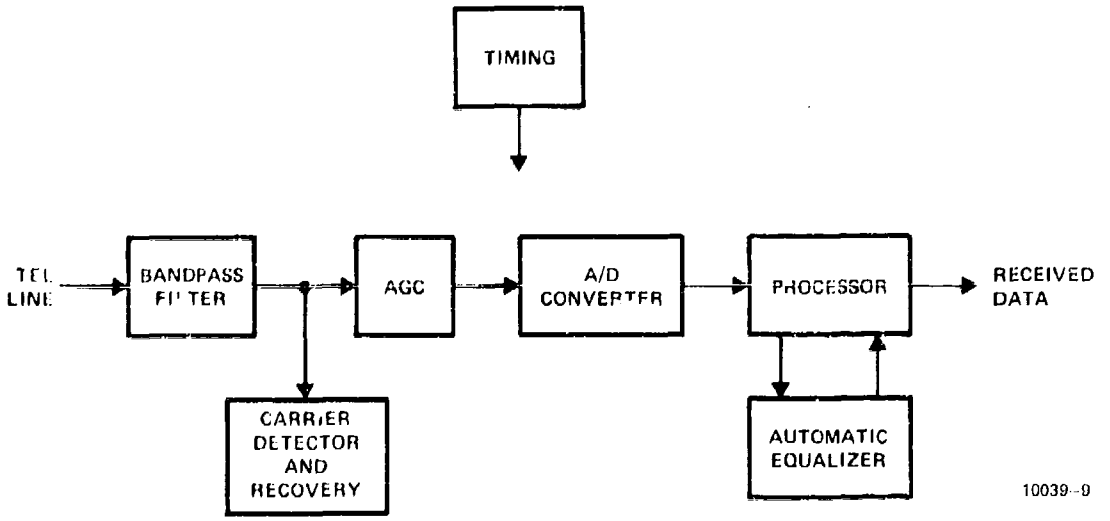


Figure 10. 16 kb/s Modem Receiver With Processor

First, add an additional processor at an additional cost of 50 IC's with power supply loading and size about the same as the present box; second, use custom LSI to reduce processor load; third, use PROM lookup tables for some functions; fourth, use the multiplier in the automatic equalizer to speed up the modulator/demodulator function; fifth, speed up the processor's microcycle; sixth, a combination of the above. Another problem is the amount of engineering time required to implement the various changes. This can be reduced with the use of the proper development system but will still require a major effort, since, due to the severe speed constraints, the software for the procesor must be very efficient. That is, the number of functions that are able to be accomplished in the processor will be directly related to the quality of software.

2.11 Software Development

The hardware needed and the processor time required to implement the various functions as previously discussed is based upon the processor software being microprogrammed. This is necessary in order to obtain the maximum processing power during each clock period or microcycle. At this level operations are referred to as microinstructions and the information is stored in the control memory as microcode. With microprogrammed machines we are able to control all areas of the processor independently during each microcycle. That is the various areas of the processor (CPU, controller, I/O, data memory and external modules) can be programmed as desired for maximum performance during each microinstruction. This is basically a parallel processor and as such obtains its power from the use of wide (56 bits) microinstructions. With this many bits for each microinstruction it is easy to see why each microinstruction can perform a tremendous number of different operations. By taking advantage of this we are able to customize each microinstruction for maximum efficiency from the processor. Combining this with the fact that the processor uses bipolar circuits we arrive at the fastest solution for a given problem. Even with a processor this powerful many microinstructions are necessary to perform the functions of the 16 kb/s modem. It is estimated that one thousand (1000) lines of microcode will be required to perform these

functions and an additional five hundred (500) lines for debug and diagnostics aids. Thus a total of 1500 microinstructions, 56 bits wide, must be written and debugged. In addition, great care must be taken when writing the code to optimize it if all the desired functions are to be accomplished in the available time. Because of this a significant amount of engineering effort will be required to develop and refine the software. Therefore many aids to speed up software development will be necessary.

The first step towards simplifying the software development is to replace the 56 bit binary microinstructions with something the programmer can more easily understand. This is done by dividing the microinstructions into fields and assigning alphanumeric names corresponding to the operations that field controls. These alphanumeric names are referred to as mnemonics. Each field describes either a function, a source register, or a destination register. Table 3 shows the fields assigned to the various bits in the microcode for the processor. The fields are usually encoded, i.e., a register file containing 16 registers will use only four bits of microcode to select the appropriate register rather than 16. A good example of a field and the mnemonics associated with it is the three bits required to control the ALU functions of the 2901. The definition of the three bits is: 000 = R + S, 001 = S - R, 010 = R - S, 011 = R or S, 100 = R and S, 101 = R-NOT and S, 110 = R EX-OR S, 111 = R EX-NOR S. In this example R and S are both source registers. If mnemonics were assigned to the different operations they might have the form: ADDRS = 000, SUBSR = 001, SUBRS = 010, ORRS = 011, ANDRS = [illegible]'NDRNOTS = 101, EXORSS = 110, EXNORRS = 111. If these mnemonics were [illegible]igned then the statement "ADDRS, R9, S2" would represent 000, 1001, 0010 in the binary number system if R and S were register files of 16 words each. Similarly the statement "ORRS, R11, S8 would represent the binary pattern 011, 1011, 1000. Since we will be writing our software in mnemonics, it would be desirable to generate a computer program to convert the mnemonics into the appropriate bit patterns. This type program is normally referred to as an assembler. However, since we are defining the fields for our particular processor rather than using a standard set (8080, or 6800, etc.), a meta-assembler is what is really

Table 3. Microcode Field Bit Assignments

Bits	Field
1 - 12	Branch address or data inputs
13	Enable branch address onto data bus
14 - 17	Sequencer control
18 - 23	Select test condition
24 - 32	2901 control bits
33 - 36	A address
37 - 40	B address
41	2901 output enable
42	2901 carry input
43 - 44	2901 shift control
45	Store status bits from 2901
46 - 49	RAM/PROM control
50 - 53	I/O control
54 - 56	Miscellaneous

required. This is an assembler that allows all the mnemonics to be defined by the user before any assembling is done. There are various meta-assemblers available for use with our system; AMDASM, Rapid, Dapel, etc. Another advantage of these assemblers is that they allow symbolic addressing. That is the programmer can use symbols to refer to locations in his program rather than line numbers. Not only does this allow labeling as to the function being performed at a given location, but if new microinstructions are added, changing line numbers, the existing references to locations whose line number have changed is unaffected since the labels are preserved. All these features aid the programmer in writing instructions in a language understandable to him with a computer converting this into binary patterns that the processor understands.

Once the software has been written and assembled it has to be loaded into the control section of the processor for debugging. This can be done by taking the assembled data and blowing PROM's which are then plugged into the control storage. However, with a software problem of our magnitude this is time consuming (there are seven 1K x 8 bit PROM's) and since the PROM's have to be of the fusible link type, each change in software will require new PROM's. These problems can be avoided if the control storage is implemented in random access memory's (RAM's) and the computer used to assemble the microcode is capable of loading its output files directly into this RAM. This now allows the programmer to debug his code in an efficient manner. Also if mass storage (floppy-disk, etc.) is available to the computer other programs can be called up as required for diagnostics. When the software is debugged and operational in RAM, then the PROM's can be blown and plugged into the processor for final system checkout. A development system of this nature, the System 29, is currently available from Advanced Micro Devices. It is a complete microprogrammed system development lab designed to support designs of our type.

2.12 Conclusions and Recommendations

An AMD 2901 bit-slice microprocessor implementation approach to redesigning the 16 kb/s modem appears feasible. The AMD 2901 Processor would replace the existing processor and allow inclusion of many outboard functions within the processor. The estimated reduction ir cost/complexity/size of the modem is 33 to 50 percent. However, a change of this nature is extensive and will require a significant development effort.

Since the study resulted in a promising approach to reducing the cost/complexity/size of the modem, it is recommended that a future effort be undertaken to better define the development approaches and cost.

3.0 HF TESTING

3.1 Background

During the 16 kb/s modem test program (Contract F30602-76-C-0460) measurements were made which involved transmission of 16 kb/s CVSD over a short HF link in Oahu. An error rate of 9.8×10^{-3} was achieved in this test which provided encouragement to the potential of the modem in possible HF applications. Since the problems associated with long haul HF links were anticipated to be more severe on modem operation, a brief long haul HF test was planned as a part of the partitioning study (Contract F30602-77-C-0190). The objective of this test was to assess the nature of modem modifications that would be necessary to allow operation over long haul HF link.

3.2 Site Survey

The sites which were selected were Malabar, Florida, which serves as the transmitting site for the Cape Kennedy HF support activities, and Ava, New York, located near Rome, New York. A site survey was conducted prior to testing. During the site survey an attempt was made to establish a ground-wave loop-around from the technical control site at Cape Kennedy via telephone to Malabar and via HF back to Cape Kennedy. The error rates in this link were in the order of 6 percent even though a relatively stable received signal level was obtained. By making further measurements at the transmitter site at Malabar, it was discovered that the problem was caused by high frequency phase jitter which was present in the Collins 310 F6 exciter associated with the 204C transmitter that was used in the loop tests. The phase characteristics of several other 310 exciter were measured, and it was found that the jitter varied considerably depending upon the particular exciter and particular frequency being tested. However, transmitters utilizing the 310 V1 exciter had a relatively small amount of low frequency phase jitter over a wide range of frequencies. Table 4 lists the phase jitters measured on these units.

Table 4. Measured Phase Jitter on Exciters at Malabar Transmit Site

Exciter Type	Exciter No	RF Freq MHz	P-P Phase Jitter^{0}	Freq. of Jitter (Approx.) Hz
310F6	Test	12.105	20^{0}	180
		15.000	23^{0}	180
		30.000	50^{0}	180
		25.000	46^{0}	180
		20.000	41^{0}	180
		16.000	40^{0}	180
		4.000	10^{0}	180
		7.560	12^{0}	180
		3.600	6^{0}	180
		2.500	4.5^{0}	180
310F6	C7	9.043	8^{0}	120*
		12.105	20^{0}	120
		18.384	11^{0}	120
		30.300	55^{0}	120
		7.350	Large Phase Hits	- -
		5.000	3^{0}	120
310F6	C6	12.105	13^{0}	Random Noise
		15.000	16^{0}	Random Noise
		30.300	30^{0}	Random Noise
310V1	Spare	10.192	7^{0}	16
		25.000	17^{0}	16
		12.105	8^{0}	16
		4.000	3^{0}	16
		15.000	12^{0}	16

* Phase Waveform had spikes of approximately 8 ms width at a 120 Hz rate.

Receivers located at the Cape Kennedy receive site were also tested relative to phase jitter characteristics as a function of frequency, and harmonic distortion readings as a function of input signal level and frequency. Test results from these different receivers are presented in Table 5. The 651F1 receivers use the same frequency synthesizer as the 310 F6 exciter tested at Malabar and the 50 E6 uses the same synthesizer as the 310 V1 tested at Malabar. The distortion analyzer readings were obtained using a 1 kHz tone and a harmonic distortion analyzer that measures the ratio of total signal with the 1 kHz tone notched out to total signal without the notch. Hence, it reads harmonic distortions at high signal strength and signal-to-noise at low strength. The general conclusion reached from these measurements were that the transmitters and receivers using the synthesizer in the 310 V1 exciter were capable of supporting the 16 kb/s signal without severe degradation. However, those using the 310 F6 synthesizer were not satisfactory at any choice of RF frequency.

Since it was not possible to test the actual receivers to be used at the AVA test site prior to conducting the tests, a Harris RF 550 receiver was obtained and tested with the 16 kb/s modem. This was accomplished by using a balanced mixer as the transmitter. The bit error rates measured through the recevier were generally below 10^{-3}. The principal cause of errors at high signal strength appeared to be harmonic distortion associated with the AGC circuitry since the modem ran nearly error free when manual AGC was used. This receiver was taken to Ava for the HF tests.

3.3 Test Results

The long haul HF tests were conducted in October 1977. The frequencies assigned for us during the test were 2724 kHz, 8116.5 kHz, 10648 kHz, 12165 kHz, 16140 kHz and 18535 kHz. Some difficulty was experienced in obtaining the desired equipment and frequencies due to a delayed launch at Cape Kennedy. However, enough tests were conducted to obtain the basic information that was sought. The results were that the modem could not achieve initial equalization on any of the long haul

Table 5

Receiver Type	No	RF Freq. (MHz)	Phase Jitter	Input Signal	Dist.
651F1	1	12.105	2^{o}	1 mV	-41 dB
				2.2 μV	-27 dB
				1 μV	-21 dB
		29.000	3^{o}	300 μV	-36 dB
		4.000	1.5^{o}	10 μV	-20 dB
651F1	2	4.000	1.5^{o}	10 μV	-38 dB
				1 μV	-20.5 dB
		12.105	1.6^{o}	10 μV	-38 dB
				21 μV	-21 dB
		29.000	4.5^{o}	30 μV	-36 dB
				10 μV	-31.5 dB
				1 μV	-13 dB
50E6		29.000	8.5^{o}	1 mV	-20 dB
				100 μV	-18 dB
				30 μV	-10 dB
		12.105	4.5^{o}	1 mV	-27 dB
				100 μV	-20 dB
		4.000	20^{o} Jumps	1 mV	-23 dB

circuits tested. Attempts were made on most of the assigned frequencies including some very close to the maximum usable frequency (MUF) for that day. In all cases the received signal strength was good so that receiver noise should not have been a problem. Bit error rate tests were conducted through the receiver (Type RF551F-1) at AVA as well as through the RF550 receiver by using the balanced mixer as a transmitter. Error rate values between 10^{-2} and 10^{-3} were achieved. Thus it was clear that the dispersive dynamic multipath associated with the propagation link was the difficulty. This also could be verified by observing the equalizer weights during training since they clearly were attempting to track variations beyond their speed capability.

To obtain a more quantitative measure of the dynamic channel dispersion an experiment was arranged involving transmission of four tones spaced through the audio band. These tones were separately filtered and supplied to a six channel recorder. A typical segment of these recordings is shown in Figure II. Although a large amount of recordings was obtained at various carrier frequencies, the statistical behaviour exhibited in Figure II is essentially representative of all of them. The lowest channel of the recording shown in Figure 11 is the AGC voltage. Although calibration is not shown, the signal-to-noise ratio during the entire period of the recording is sufficient to allow 16 kb/s data transmissions. The top trace is the sum of the four in-band sine waves which can be observed as being maintained at a relatively constant value by the receiver AGC. The middle four channels are the outputs of the four filters at 3.2 kHz, 1.8 kHz, 1 kHz and 600 kHz from bottom to top. The vertical lines represent 1 second intervals. The significant features to note are the severe fades that occur on individual frequency components at a rate of approximately once per 4 seconds. These are obviously caused by two reflected paths that are nearly equal in strength with the relative path length varying at a nearly constant rate such that an increase or decrease of a wavelength of the carrier frequency takes approximately 4 seconds. In addition, a smaller reflection with path length varying at approximately four times that rate is present as evidenced by the smaller part fluctuations. Since the carrier frequency was 10648 kHz a wavelength

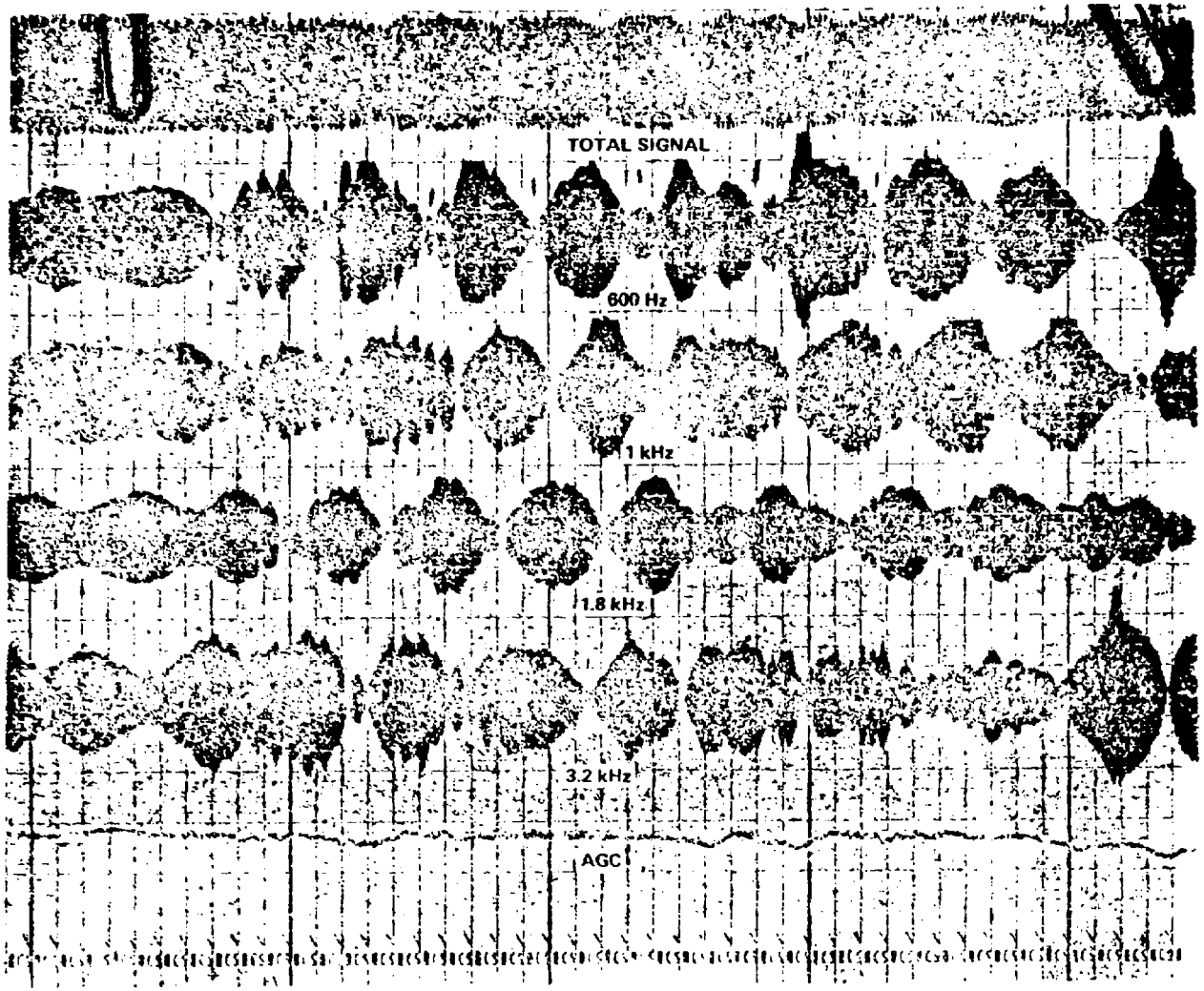


Figure 11. Typical Recording From HF Tests

is approximately 30 meters. In addition the relative size of the components is clearly also varying with time since the null depth also varies. These observations are quite consistent with channel models such as that described by Watterson, Jurocher and Bensema.[4]

3.4 Conclusions From HF Tests

Several conclusions can be drawn from the site survey and HF Tests.

1. The 16 kb/s modem in its present form (or probably in any other form) is not likely to be capable of operating satisfactorily over the complete inventory of HF receivers and transmitters even in a back-to-back mode. On the other hand there are existing transmitters and receivers that are acceptable.

2. The four tone test conducted in the tests indicates that an acceptable equalizer for combatting dispersion in an HF channel will need to have the capability of tracking its weights at rates of several Hz which is considerably faster than the existing wire line design can handle.

3. In addition to the above problem, the presence of frequently occurring in-band nulls indicates that the type equalizer being used for the wire line application is the wrong type to use for HF. This occurs since the wire line modem equalizer synthesizes an inverse filter rather than a direct channel filter. The inverse filter of a channel with nulls at some frequency require either very large weight values or very long length.

4. Because of the previous two conclusions it is felt that additional study will be required to determine the best manner to modify the wire line recorder for HF application.

REFERENCES

1. "16 kb/s Data Modem," Report F30602-75-C-0129, 14 June 1976, Rome Air Development Center, DCLD, Griffiss Air Force Base, N.Y. 13441, B015627L.

2. "16 Kilobit Modem Evaluation," Report F30602-76-C-0460, 27 July 1977, Rome Air Development Center, DCLD, Griffiss Air Force Base, N.Y. 13441, B029131L.

3. Logan, H. L., Jr. and G. D. Forney, Jr., "A MOS/LSI Multiple Configuration 9600 BPS Data Modem," Proceedings of the ICC 1976, pp. 48.7-48.12, June 1976.

4. Waterson, C. C., J. R. Jurosher, W. D. Benslina, "Experimental Confirmation of an HF Channel Model," IEEE Transactions on Communications, pp. 792-803, Dec. 1970.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.

DEPARTMENT OF THE AIR FORCE
AIR FORCE RESEARCH LABORATORY (AFMC)

1 Jun 04

MEMORANDUM FOR DTIC-OCQ
ATTN: Larry Downing
Ft. Belvoir, VA 22060-6218

FROM: AFRL/IFOIP

SUBJECT: Distribution Statement Change

1. The following documents have been reviewed and have been approved for Public Release; Distribution Unlimited:

ADB084552, "Project Birdwatch at Dover AFB", RADC-TR-84-7

ADB191869, "Acousto-Optic Beam Steering Study", RL-TR-94-121

AD0800669, "Use of Commercial Broadcast Facilities for Emergency DoD Communications", RADC-TR-66-392

ADB058979, "Multi-Rate Secure Processor Terminal Architecture Study", RADC-TR-81-77, Vol 1.

ADB053656, "16 KB/S Modem (AN/GCS-38) CONUS Test", RADC-TR-80-89

ADB055136, "VINSON/AUTOVON Interface Applique for the Modem, Digital Data, AN/GCS-8", RADC-TR-80-341

ADB043556, "16 KB/S Data Modem Partitioning", RADC-TR-79-278

ADB029131, "16 Kilobit Modem Evaluation", RADC-TR-78-127.

2. Please contact the undersigned should you have any questions regarding this document. Thank you very much for your time and attention to this matter.

KELLY CAGWIN
STINFO Officer
Information Directorate
315-330-7094/DSN 587-7094